500.36133CC2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: SHIMOKAWA et al.

Serial No.: Rule 53(b) of 09/044,163

Filed: February 28, 2002

For: DATA STRUCTURE IN DATABASE, DATABASE SYSTEM FOR MANAGING DATABASE AND DATABASE AND DATABASE MANAGING METHOD AND SYSTEM

Art Unit: Unassigned (2171 previously in parent application)

Examiner: Unassigned (U. Le previously in parent application)

PRELIMINARY AMENDMENT

Commissioner for Patents
Washington, D.C. 20231

February 28, 2002

Sir:

IN THE SPECIFICATION

Please amend the specification as follows:

Page 1, before the first paragraph, insert the following sentence:

--This is a continuation of parent application Serial No. 09/044,163, filed March 19, 1998.--

Please replace the paragraph beginning at page 1, line 2, with the following rewritten paragraph:

--The present invention relates to a time series database processing system, of an especially ultra-large scale, for storing data pieces serving as updating

detailed information in a sequence of time series in a database and for controlling addition/deletion/retrieval of data.--

Please replace the paragraph beginning at page 1, line 7, with the following rewritten paragraph:

--When data pieces are loaded on a database of a large scale and a specified data piece is retrieved from the database, an index is generally applied. Indexing is effective when an item serving as a key during retrieval can be specified. The indexing is a contrivance in which specified key items of a database are collected, a pointer is provided over the key items to take the form of a balanced tree (B tree), and the tree can be traced at a high speed up to a location corresponding to a leaf of the tree in accordance with information indicating which range a key of a specified value lies in. "An Introduction to Database Systems, 3.4 Indexing" by C. J. Date, Addison-Wesley, 1986, pp. 68-77 teaches a contrivance in which information corresponding to storage locations of all data items can be obtained for all the data items. If the database is for about million cases or events, there occurs no problem. But in a database of an ultra-large scale for billion cases or trillion cases, however, the maintenance of index per se swells, and especially, keys which are added in time series fashion may not be handled well.--

Please replace the paragraph beginning at page 2, line 19, with the following rewritten paragraph:

--A utility for data loading uses a technique for writing data directly to a physical area of a database and therefore, with this utility, data can be written at a

high speed. However, the utility for high-speed data loading generally inhibits direct data writing to the physical area during data loading from a conflicting area at other retrieval or updating access. In other words, data loading shall compulsorily be executed while inhibiting access to a specified table for retrieval/updating or a part of a table for retrieval/updating. This forces retrieval of the database to be once stopped each time that time series data is loaded, which can be on a daily basis. In a database of ultra-large scale, it takes one day or more for retrieval per se in some applications. In that case, data loading cannot be permitted unless retrieval is stopped, leading to fatal inconvenience. To avoid such situations, data can be added through usual data insertion operation without resorting to data loading, but in this case the performance is degraded by approximately by one order as compared to data loading of a physical writing type. Besides, locking must be acquired for concealing data during addition, largely affecting the performance of operation for retrieval of all cases or events in the database.--

Please replace the paragraph beginning at page 3, line 15, with the following rewritten paragraph:

--In order to delete a data piece in the database for which a constant time has expired, the data piece is typically required to be retrieved, and even in the case of an index, the time consumed in comparison to that for inserting data piece by piece is significant. In the absence of index, all data pieces are retrieved for the purpose of deleting a data piece of interest and consequently, in the database of ultra-large scale, it takes one day or more to operate only the deletion processing and practically, the time series database cannot be materialized.--

Please replace the paragraph beginning at page 3, line 25, with the following rewritten paragraph:

--Thus, for the deletion of data for which a constant time has expired, time exceeding that for retrieval of all pieces of data is consumed in the absence of an index but conversely, in the presence of an index, indexing is updated during deletion, leading to an operation which consumes much time as in the case of data insertion. Accordingly, it is practically difficult to realize daily data deletion for the database which takes one day or more to retrieve all data pieces.--

Please replace the paragraph beginning at page 4, line 7, with the following rewritten paragraph:

--An object of the invention is to provide a method and system which can eliminate conflict of the operation of time series data loading and data deletion with the operation of data retrieval in a database system and which can mitigate suppression imposed on retrieval by the system.--

Please replace the paragraph beginning at page 4, line 13, with the following rewritten paragraph:

--Another object of the invention is to provide a database managing system which can dispense with reorganization of an index tree which loses balance due to the addition of time series data.--

Please replace the paragraph beginning at page 5, line 5, with the following rewritten paragraph:

--loading time series data pieces for the predetermined time in a plurality of data areas in the database at a sequence of times corresponding to the time series data pieces.--

Please replace the paragraph beginning at page 5, line 15, with the following rewritten paragraph:

--detecting the bookmark information including the designated time, and, when the state transition information included in the detected bookmark information indicates the online state, reading a time series data piece corresponding to the detected bookmark information.--

Please replace the paragraph beginning at page 5, line 20, with the following rewritten paragraph:

--When the state transition information included in the detected bookmark information indicates either a value indicative of the loading state or a value indicative of the empty state, it can be determined that the data retrieval request has not yet been responded to.--

Please replace the paragraph beginning at page 6, line 3, with the following rewritten paragraph:

--detecting the bookmark information including the designated time, and, when the state transition information included in the detected bookmark information indicates the online state, setting a value indicative of the empty state in the state transition information included in the detected bookmark information.--

Please replace the paragraph beginning at page 6, line 10, with the following rewritten paragraph:

--cumulating repeatedly applied time series data pieces in a cumulative data storage area until the cumulative data reaches a total data for the predetermined time; and--

Please replace the paragraph beginning at page 7, line 16, with the following rewritten paragraph:

--In the present invention, the database is divided into segments which are each minimum blocks for storage area management and time series data pieces which are stored in the segments. When data is loaded on the database, a time at which the data is loaded is stored as a bookmark at a predetermined location in a start segment from which the addition starts with the database. Thanks to the bookmark, when retrieval of time designation or time interval designation is carried out, the retrieval range can be narrowed physically by utilizing the bookmark.--

Please replace the paragraph beginning at page 7, line 26, with the following rewritten paragraph:

--When data loading is effected, the database can be brought into a loading unfinished state by locating the bookmark in other places than the place in which the data is being loaded. Consequently, data can be loaded directly on a physical segment without affecting other retrieval. At the time that the data loading is completed, the bookmark is written in the above other place and the database is recognized by such assigning a bookmark thereto.--

Please replace the paragraph beginning at page 8, line 7, with the following rewritten paragraph:

--In the case of data deletion, when data pieces following a specified bookmark are to be deleted collectively, the areas are effectively emptied changing the bookmark for the unit of segment within a short time without actually accessing [to] the data. By managing the areas of the database in a unit of segment in wrap-around fashion, the always pooled consecutive areas can be used from one side to load data and replenish an area from the other side of the consecutive areas.--

Please replace the paragraph beginning at page 10, line 19, with the following rewritten paragraph:

--Referring to Fig. 3, the construction of the 20 storage apparatus 13 is shown in greater detail to give a detailed explanation of the system definition information area 15 and data area 14. In the present embodiment, the data area 14 has consecutive areas secured on the storage apparatus 13 so as to be divided into management blocks called segments 20. Data pieces generated in time series fashion are put together in the area 8 by means of the management program 12 until they reach an amount for a constant time. The collected data pieces are stored in one of the management block segments of the consecutive areas of the database in the form of the storage apparatus 13, along with a time for storage which is read out of a clock 9 and stored in the same segment or otherwise at a different location. The segment 20 includes, for example, a data storage area 21 for storing real data and a bookmark information area 22 for storing management information for the

data stored in the data storage area 21. In the present embodiment, the segment 20 consists of a plurality of pages, each being a unit of disk input/output.--

Please replace the paragraph beginning at page 11, line 12, with the following rewritten paragraph:

--The system definition information 15 has information for managing the storage location of time series data, including information for pointing to a segment 20 which is the oldest in time series and information for pointing to the start of an empty segment area.--

Please replace the paragraph beginning at page 12, line 6, with the following rewritten paragraph:

--In the time series database, retrieval for which time is specified is frequently practiced. For example, the title and the date of issue of a book published by a publisher are stored in time series fashion in a time series database of the publisher by using the issue date as a key, and an example will be described hereunder in which the database is retrieved for a list of titles of books issued over three months which range from March, 1994 to May, 1994.--

Please replace the paragraph beginning at page 12, line 21, with the following rewritten paragraph:

--In the database system of the present embodiment, information for pointing to a segment 20 which stores the oldest data in time series is first acquired from the system definition information 15 (step 500). Then, the database system acquires

time information t (February, 1994) and status information (online) from a bookmark information area 22 of the pointed segment 20 (step 501). Acquisition of the system definition information is carried out at a high speed because a predetermined capacity of data can be acquired starting with the start of a plurality of segments arrayed at equi-capacity intervals on the database.--

Please replace the paragraph beginning at page 13, line 10, with the following rewritten paragraph:

--If the status information is "online", access is permitted and the program proceeds to the next process (step 503). The posterior retrieval request time (May, 1994) is compared with the time information (February, 1994) stored in the bookmark information area 22 to decide whether the intended data is stored in the database. If the result of comparison is "Yes", in a test to determine whether the stored newer data is newer than the range of the retrieval object (March, 1994 to Nay, 1994), the retrieval processing ends. When "No" is issued in the decision process, the program proceeds to the next process (step 504) to decide whether the segment 20 now pointed to is within the retrieval request time (March, 1994 to Nay, 1994). Since the segment 20 is of February, 1994, this data storage area 21 is excluded from the retrieval object and a segment 20 for storing data which succeeds in terms of time series is pointed to (step 506). For example, it is assumed that a magnetic disk device is used as the storage apparatus 13 and given that all of the segments 20 have the same size, the succeeding segment can be pointed to by moving the size of segment (a moving amount relative to the magnetic head) starting from the header of the present disk.--

Please replace the paragraph beginning at page 14, line 5, with the following rewritten paragraph:

--Next, for that succeeding segment 20, the decision process similar to the above (steps 502, 503 and 504) is executed. When it is determined in the process (step 504) that the segment 20 is one which meets the retrieval request, data is read out of the corresponding data storage area 21 in the segment 20 (step 505). Since the header of the disk points to the start of a segment 20 which stores the next data in terms of time series after the data has been read out of the data storage area 21 (step 506), time information is again acquired from a bookmark information area 22 and thereafter, the decision is repeated in a similar way. In this manner, the segments 20 are sequentially read. Since in the decision process (step 503) of a segment 20 the segment is determined to be outside the retrieval object, the retrieval processing ends at that time.--

Please replace the paragraph beginning at page 15, line 4, with the following rewritten paragraph:

--Firstly, empty segment information is read out of 5 the system definition information 15 (step 600). An empty segment 20 is pointed to by that information. In order to read input data, the input file is accessed and data (July, 1994) is read (step 602). Because of the presence of the data, "presence" is determined in the process (step 602) and the program proceeds to the process (step 603). In the process (step 603), a write process is executed. Firstly, the time, information (July, 1994) is written at the time information area and a flag "loading" indicating, currently loading at the status flag area in the bookmark information area 22, and data is

written into the data storage area 21. After completion of the data writing, a state as shown in Fig. 8 prevails.--

Please replace the paragraph beginning at page 15, line 18, with the following rewritten paragraph:

--After the writing of data for one segment has been terminated, the database system reads the next input data from the file (step 601). Because of the presence of data for August, 1994, "presence" is determined in the decision process (step 602). Through the same logic as that used for writing the data for July, 1994, time information (August, 1994), a status flag "loading" and data are written at the time information area 23, status flag area 24 and data storage area 21 in a segment 20 (step 603).--

Please replace the paragraph beginning at page 15, line 27, with the following rewritten paragraph:

--After completion of the data writing, the system is about to read the next data from the file (step 601). But, since data has already been absent in the file, "absence" is determined in the decision process (step 602) and the program proceeds to the next process (step 604 in Fig. 7).--

Please replace the paragraph beginning at page 18, line 3, with the following rewritten paragraph:

--A segment 20 which is next in terms of time series is pointed to (step 705) and time information (March, 5 1994) is acquired from the bookmark information

area 22 of that segment 20. The acquired time information (March, 1994) is compared with February, 1994 for the deletion object and it is determined that the segment 20 is not the deletion object (step 701), thus ending the deletion processing. After the completion, the database assumes a state as shown in Fig. 11.--

Please replace the paragraph beginning at page 18, line 12, with the following rewritten paragraph:

--In the present deletion processing, internal data need not be directly accessed and only the bookmark information area is taken as the object, thereby making it possible to perform deletion within a short time and while online.--

Please replace the paragraph beginning at page 18, line 17, with the following rewritten paragraph:

--The segments are used in wrap-around fashion to attain an advantage in that no reorganization is needed even when addition/deletion is repeated. Finally, the wrap-around architecture will be described.--

Please replace the paragraph beginning at page 19, line 22, with the following rewritten paragraph:

--As described above, according to the embodiments of the present invention, the intended data can be accessed without resorting to an index by retrieving thoroughly only the specified control information storage range without retrieving the whole of the database.--

IN THE CLAIMS

Please cancel original claims 1-20 without prejudice or disclaimer.

Please add the following new claims:

--21. A data structure, stored on a storage medium, in a database, comprising:

a plurality of data areas, each of said plurality of data areas being loaded with data each for a constant time generated in time series during a certain time, the plurality of data areas being managed by the time series; and

bookmark information areas respectively provided at predetermined locations in said plurality of data areas, each having a pair of bookmark information indicative of a time at which said data is loaded in a time series data piece for said constant time in each of said data areas and state transition information indicative of a state of the data piece in said each data area, said state transition information being allowed to have one of a value indicative of an online state in which the data area is permitted to be retrieved and a value indicative of a loading state in which loading of data in the data area has not yet been completed and the data area is not permitted to be retrieved.

22. A data structure, stored on a storage medium, in a database, comprising:

a plurality of data areas in which given time series data pieces each for a constant time are loaded at predetermined locations, respectively, in said database, each of said plurality of data areas being loaded with data generated in time series during a certain time, the plurality of data areas being managed by the time series; and

predetermined bookmark information areas each having a pair of bookmark information indicative of a time at which said data is loaded in a time series data piece in each of said data areas and state transition information indicative of a state of the data piece in each data area, said state transition information having one of a value indicative of an online state in which the data area is permitted to be retrieved and a value indicative of a loading state in which loading of data in each data area has not yet been completed and the data area is not permitted to be retrieved.

23. A database managing method for managing data in a database, comprising:

adding, to a predetermined location in a given time series data piece for a predetermined constant time, bookmark information having bookmark information indicative of a time at which said data is loaded in a time series data piece for said predetermined constant time for said predetermined time and state transition information indicative of a state of said time series data piece for said predetermined constant time;

providing, as said state transition information, one of a value indicative of an online state in which the data area is permitted to be retrieved, a value indicative of a loading state in which loading of data in the data area has not yet been completed and the data area is not permitted to be retrieved and a value indicative of a state in which data in the data area is empty; and

loading time series data pieces for predetermined constant times in a plurality of data areas in said database, each of said plurality of data areas being loaded with

data generated in time series during a certain time, the plurality of data areas being managed by the time series.

24. A database managing method for managing data in a database, comprising:

adding, to a predetermined location in a given time series data piece for a predetermined constant time, bookmark information having bookmark information indicative of a time at which data is loaded in a time series data piece for said predetermined constant time and state transition information indicative of a state of said time series data piece for said predetermined constant time and start area information having a flag indicating whether the area is the final one of a plurality of areas in said database and an address area for setting an address;

providing, as said state transition information, one of a value indicative of an online state in which the data area is permitted to be retrieved and a value indicative of a loading state in which loading of data in the data area has not yet been completed and the data area is not permitted to be retrieved;

loading time series data pieces for predetermined constant times in a plurality of consecutive data areas in said database, each of said plurality of consecutive data areas being loaded with data generated in time series during a certain time, the plurality of consecutive data areas being managed by the time series; and

raising said flag of start area information in the final one of said plurality of consecutive data areas and setting an address of first one of said plurality of consecutive data areas in said address area.

25. A database managing method for managing data in a database, comprising:

reading bookmark information having bookmark information indicative of a time at which data is loaded in a time series data piece for a predetermined constant time and state transition information indicative of a state of said time series data piece for said predetermined constant time from a predetermined bookmark area and setting the state of said time series data piece in said state transition information to a value indicative of a state in which data is empty so as to write said bookmark information in said database; and

loading given time series data pieces for given predetermined constant times in a plurality of data areas in said database, each of said plurality of data areas being loaded with data generated in time series during a certain time, the plurality of data areas being managed by the time series; and

writing bookmark information having bookmark information indicative of a time corresponding to a time series data piece for said predetermined constant time and state transition information indicative of an online state of said time series data piece for said predetermined time in said predetermined bookmark area.

26. A database managing method according to claim 25, further comprising:

cumulating repeatedly applied time series data pieces in a cumulative storage area until the cumulative data reach total data for said predetermined constant time; and

adding, to a data piece in said cumulative data storage area, bookmark information having bookmark information indicative of a time at which said data is

loaded in said data piece for said predetermined constant time and state transition information indicate of a state of said time series data piece for said predetermined constant time and loading resulting data pieces in said plurality of data areas in said database, each of said plurality of data areas being loaded with data generated in time series during a certain time, the plurality of data areas being managed by the time series.

27. A database managing system, comprising:

a processor having a memory for storing data for a certain time and a clock for reading times at which said data are applied, the data in the memory being managed by time series; and

a database connected to said processor and having bookmark information indicative of a time at which said data is loaded in a time series data piece for a predetermined constant time, state transition information indicative of a state of said time series data piece of said predetermined constant time and said time series data pieces for said predetermined constant times, said state transition information having one of a value indicative of an online state in which the data area is permitted to be retrieved, a value indicative of a loading state in which loading of data in the data area has not yet been completed and the data area is not permitted to be retrieved and a value indicative of a state in which data in the data area is empty.

28. A data base managing method for managing data in a database comprising the steps of:

pointing, in response to a retrieval request requesting data for a constant time in time series between a first time and a second time, to a segment of a database which stores a data oldest in time series between said first time and said second time;

acquiring time information from a bookmark residing at a predetermined position of said segment to obtain status information to determine whether said status information indicates a state of loading of data in said database;

seeking succeeding segments to find segments of time series after said first time based on bookmarks of said succeeding segments until a segment of time series at an end time series before said second time among said segments having status of loading; and

reading data from said segments found in the seeking step.

29. A database managing system comprising:

means, in response to a retrieval request requesting data for a constant time in time series between a first time and a second time, for pointing a segment of a database which stores a data oldest in time series between said first time and said second time;

means for acquiring time information from a bookmark residing at a predetermined position of said segment to obtain status information to determine whether said status information indicates a state of loading of data in said database;

means for seeking succeeding segments to find segments of time series after said first time based on bookmarks of said succeeding segments until a segment of

time series at an end time series before said second time among said segments having status of loading; and

means for reading data from said segments found in the seeking means.--

REMARKS

SPECIFIC REFERENCE TO EARLIER FILED APPLICATION(S)

As an application in which the benefits of an earlier application are desired must contain a specific reference to the earlier filed application(s) in the first sentence of the specification (37 CFR 1.78), the specification has herein been amended to incorporate such specific reference to earlier filed application(s).

SPECIFICATION AMENDMENTS ADOPTED FROM PARENT APPLICATION

Disclosure/specification amendments made and approved within the parent application have been effected in the present application, and any spelling, idiomatic, grammatical and/or other informality noted during further review of the disclosure/specification have been corrected.

PROPOSED DRAWING CHANGES ADOPTED FROM PARENT APPLICATION

Submitted herewith under separate cover letter are copies of ones of Applicant's drawing sheets with red ink adopting drawing changes proposed and approved within the parent application. Also submitted herewith under separate cover letter are correspondingly corrected formal drawing sheets, for which approval is respectfully requested.

CLAIM FOR PRIORITY

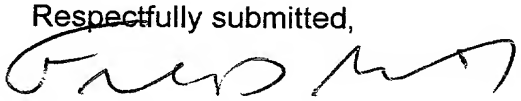
Applicant respectfully acknowledges that in order for a patent issuing on the instant application to obtain the benefit of priority under 35 USC 119(a-d) based on priority papers filed in an ancestor application, a claim for such foreign priority must be made in this application. Applicant herein (under separate cover letter) makes such claim for foreign priority, and respectfully submits that the priority papers were filed in ancestor application 09/044,163. Acknowledgment and confirmation of the perfection of Applicant's claim for foreign priority are respectfully requested.

Preliminary to the examination of the above-identified application, Applicant respectfully submits specification changes made within the ancestor application(s). Applicant herein submits the foregoing amended and added claims. It is respectfully requested that examination be performed on such claims 1, 4, 7, 12, 14 and 19-22.

Attached hereto is a marked-up version of the changes made to the claims by the current Amendment. The attached page is captioned "**Version with markings to show changes made.**"

To the extent necessary, Applicants petition for an extension of time under 37 CFR §1.136. Please charge any shortage in the fees due in connection with the filing of this paper, including extension of time fees or credit any overpayment of fees to the deposit account of Antonelli, Terry, Stout & Kraus, LLP., Deposit Account No. 01-2135 (referencing case No. 500.36133CC2).

Respectfully submitted,



Frederick D. Bailey
Registration No. 42,282
ANTONELLI, TERRY, STOUT & KRAUS, LLP

FDB/pay
(703) 312-6600

Version with markings to show changes made

IN THE SPECIFICATION

Please amend the specification as follows:

Page 1, before the first paragraph, insert the following sentence:

--This is a continuation of parent application Serial No. 09/044,163, filed March 19, 1998.--

Please replace the paragraph beginning at page 1, line 2, with the following rewritten paragraph:

--The present invention relates to a time series database processing system, of an especially ultra-large scale, for storing data pieces serving as updating detailed information in a sequence of time series in a database and for controlling addition/deletion/retrieval of data.--

Please replace the paragraph beginning at page 1, line 7, with the following rewritten paragraph:

--When data pieces are loaded on a database of a large scale and a specified data piece is retrieved from the database, an index is generally applied. Indexing is effective when an item serving as a key during retrieval can be specified. The indexing is a contrivance in which specified key items of a database are collected, a pointer is provided over the key items to take the form of a balanced tree (B tree), and the tree can be traced at a high speed up to a location corresponding to a leaf of the tree in accordance with information indicating which

range a key of a specified value lies in. "An Introduction to Database Systems, 3.4 Indexing" by C. J. Date, Addison-Wesley, 1986, pp. 68-77 teaches a contrivance in which information corresponding to storage locations of all data items can be obtained for all the data items. If the database is for about million cases or events, there occurs no problem. But [but] in a database of an ultra-large scale for billion cases or trillion cases, however, the maintenance of index per se swells, and especially, keys which are added in time series fashion may not be handled well.--

Please replace the paragraph beginning at page 2, line 19, with the following rewritten paragraph:

--A utility for data loading uses a technique for writing data directly to a physical area of a database and therefore, with this utility, data can be written at a high speed. However, the utility for high-speed data loading generally inhibits direct data [write] writing to the physical area during data loading from a conflicting area at other retrieval or updating access. In other words, data [load] loading shall compulsorily be executed while inhibiting access to a specified table for retrieval/updating or a part of a table for retrieval/updating. This forces retrieval of the database to be once stopped each time that time series data [added almost every day] is loaded, which can be on a daily basis. In a database of ultra-large scale, it takes one day or more for retrieval per se in some applications. In that case, data loading cannot be permitted unless retrieval is stopped, leading to fatal inconvenience. To avoid such situations, data can be added through usual data insertion operation without [resort] resorting to data loading, but in this case[,] the performance is degraded by approximately by one order as compared to data

loading of a physical [write] writing type [and besides,] . Besides, locking must be acquired for concealing data during addition, largely affecting the performance of operation for retrieval of all cases or events in the database.--

Please replace the paragraph beginning at page 3, line 15, with the following rewritten paragraph:

--In order to delete a data piece in the database for which a constant time has expired, the data piece is typically required to be retrieved, and even in the [presence] case of an index, the time [comparable] consumed in comparison to that for inserting data piece by piece is [consumed] significant. In the absence of index, all data pieces are retrieved for the purpose of deleting a data piece of interest and consequently, in the database of ultra-large scale, it takes one day or more to operate only the deletion processing and practically, the time series database cannot be materialized.--

Please replace the paragraph beginning at page 3, line 25, with the following rewritten paragraph:

--Thus, for the deletion of data for which a constant time has expired, time exceeding that for retrieval of all pieces of data is consumed in the absence of an index but conversely, in the presence of an index, indexing is updated during deletion, leading to an operation which consumes much time as in the case of data insertion. Accordingly, it is practically difficult to realize daily data deletion for the database which takes one day or more to retrieve all data pieces.--

Please replace the paragraph beginning at page 4, line 7, with the following rewritten paragraph:

--An object of the invention is to provide a method and system which can eliminate conflict of the operation of time series data loading and data deletion with the operation of data retrieval in a database system and which can mitigate suppression imposed on retrieval by the system.--

Please replace the paragraph beginning at page 4, line 13, with the following rewritten paragraph:

--Another object of the invention is to provide a database managing system which can dispense with reorganization of an index tree which loses balance [owing] due to the addition of time series data.--

Please replace the paragraph beginning at page 5, line 5, with the following rewritten paragraph:

--loading time series data pieces for the predetermined time in a plurality of data areas in the database [in] at a sequence of times corresponding to the time series data pieces.--

Please replace the paragraph beginning at page 5, line 15, with the following rewritten paragraph:

--detecting the bookmark information including the designated time, and, when the state transition information included in the detected bookmark information

indicates the online state, reading a time series data piece corresponding to the detected bookmark information.--

Please replace the paragraph beginning at page 5, line 20, with the following rewritten paragraph:

--When the state transition information included in the detected bookmark information indicates either a value indicative of the loading state or a value indicative of the empty state, it can be [decided] determined that the data retrieval request [is] has not yet been responded to.--

Please replace the paragraph beginning at page 6, line 3, with the following rewritten paragraph:

--detecting the bookmark information including the designated time, and, when the state transition information included in the detected bookmark information indicates the online state, setting a value indicative of the empty state in the state transition information included in the detected bookmark information.--

Please replace the paragraph beginning at page 6, line 10, with the following rewritten paragraph:

--cumulating repeatedly applied time series data pieces in a cumulative data storage area until [they reach] the cumulative data reaches a total data for the predetermined time; and--

Please replace the paragraph beginning at page 7, line 16, with the following rewritten paragraph:

--In the present invention, the database is divided into segments which are each minimum blocks for storage area management and time series data pieces which are stored in the segments. When data is loaded on the database, a time at which the data is loaded is stored as a bookmark at a predetermined location in a start segment from which the addition starts with the database. Thanks to the bookmark, when retrieval of time designation or time interval designation is carried out, the retrieval range can be narrowed physically by utilizing the bookmark.--

Please replace the paragraph beginning at page 7, line 26, with the following rewritten paragraph:

--When data loading is effected, the database can be brought into a loading unfinished state by locating the bookmark in other [place] places than the place in which the data is being loaded. Consequently, data can be loaded directly on a physical segment without affecting other retrieval. At the time that the data loading is completed, the bookmark is written in the above other place and the database is recognized by such assigning a bookmark thereto.--

Please replace the paragraph beginning at page 8, line 7, with the following rewritten paragraph:

--In the case of data deletion, when data pieces following a specified bookmark are to be deleted collectively, the areas are effectively emptied changing the bookmark for the unit of segment within a short time without actually accessing

[to] the data. By managing the areas of the database in a unit of segment in wrap-around fashion, the always pooled consecutive areas can be used from one side to load data and replenish an area from the other side of the consecutive areas.--

Please replace the paragraph beginning at page 10, line 19, with the following rewritten paragraph:

--Referring to Fig. 3, the construction of the 20 storage apparatus 13 is shown in greater detail to give a detailed explanation of the system definition information area 15 and data area 14. In the present embodiment, the data area 14 has consecutive areas secured on the storage apparatus 13 so as to be divided into management blocks called segments 20. Data pieces generated in time series fashion are put together in the area 8 by means of the management program 12 until they reach an amount for a constant time. The collected data pieces are stored in one of the management block segments of the consecutive areas of the database in the form of the storage apparatus 13, along with a time for storage which is read out of a clock 9 and stored in the same segment or otherwise at a different location. The segment 20 includes, for example, a data storage area 21 for storing real data and a bookmark information area 22 for storing management information for the data stored in the data storage area 21. In the present embodiment, the segment 20 consists of a plurality of pages, each being a unit of disk input/output.--

Please replace the paragraph beginning at page 11, line 12, with the following rewritten paragraph:

--The system definition information 15 has information for managing the storage location of time series data, including information for pointing to a segment 20 which is the oldest in time series and information for pointing to the start of an empty segment area.--

Please replace the paragraph beginning at page 12, line 6, with the following rewritten paragraph:

--In the time series database, retrieval for which time is specified is frequently practiced. For example, the title and the date of issue of a book published by a publisher are stored in time series fashion in a time series database of the publisher by using the issue date as a key, and an [instance] example will be described hereunder in which the database is retrieved for a list of titles of books issued over three months which range from March, 1994 to May, 1994.--

Please replace the paragraph beginning at page 12, line 21, with the following rewritten paragraph:

--In the database system of the present embodiment, information for pointing to a segment 20 which stores the oldest data in time series is first acquired from the system definition information 15 (step 500). Then, the database system acquires time information t (February, 1994) and status information (online) from a bookmark information area 22 of the pointed segment 20 (step 501). Acquisition of the system definition information is carried out at a high speed because a predetermined

capacity of data can be acquired starting with the start of a plurality of segments arrayed at equi-capacity intervals on the database.--

Please replace the paragraph beginning at page 13, line 10, with the following rewritten paragraph:

--If the status information is "online", access is permitted and the program proceeds to the next process (step 503). The posterior retrieval request time (May, 1994) is compared with the time information (February, 1994) stored in the bookmark information area 22 to decide whether the intended data is stored in the database. If the result of comparison is "Yes", in a test to determine whether the stored newer data is newer than the range of the retrieval object (March, 1994 to Nay, 1994), the retrieval processing ends. When "No" is issued in the decision process, the program proceeds to the next process (step 504) to decide whether the segment 20 now pointed to is within the retrieval request time (March, 1994 to Nay, 1994). Since the segment 20 is of February, 1994, this data storage area 21 is excluded from the retrieval object and a segment 20 for storing data which succeeds in terms of time series is pointed to (step 506). For example, it is assumed that a magnetic disk device is used as the storage apparatus 13 and given that all of the segments 20 have the same size, the succeeding segment can be pointed to by moving the size of segment (a moving amount relative to the magnetic head) starting from the header of the present disk.--

Please replace the paragraph beginning at page 14, line 5, with the following rewritten paragraph:

--Next, for that succeeding segment 20, the decision process [like] similar to the above (steps 502, 503 and 504) is executed. When it is determined in the process (step 504) that the segment 20 is one which meets the retrieval request, data is read out of the corresponding data storage area 21 in the segment 20 (step 505). Since the header of the disk points to the start of a segment 20 which stores the next data in terms of time series after the data has been read out of the data storage area 21 (step 506), time information is again acquired from a bookmark information area 22 and thereafter, the decision is repeated in a similar way. In this manner, the segments 20 are sequentially read. Since in the decision process (step 503) of a segment 20 the segment is determined to be outside the retrieval object, the retrieval processing ends at that time.--

Please replace the paragraph beginning at page 15, line 4, with the following rewritten paragraph:

--Firstly, empty segment information is read out of 5 the system definition information 15 (step 600). An empty segment 20 is pointed to by that information. In order to read input data, the input file is accessed and data (July, 1994) is read (step 602). Because of the presence of the data, "presence" is determined in the process (step 602) and the program proceeds to the process (step 603). In the process (step 603), a write process is executed. Firstly, the time, information (July, 1994) is written at the time information area and a flag "loading" indicating, currently loading at the status flag area in the bookmark information area 22, and data is

written into the data storage area 21. After completion of [write] the data writing, a state as shown in Fig. 8 prevails.--

Please replace the paragraph beginning at page 15, line 18, with the following rewritten paragraph:

--After [write] the writing of data for one segment has been terminated, the database system reads the next input data from the file (step 601). Because of the presence of data for August, 1994, "presence" is determined in the decision process (step 602). Through the same logic as that used for writing the data for July, 1994, time information (August, 1994), a status flag "loading" and data are written at the time information area 23, status flag area 24 and data storage area 21 in a segment 20 (step 603).--

Please replace the paragraph beginning at page 15, line 27, with the following rewritten paragraph:

--After completion of [write] the data writing, the system is about to read the next data from the file (step 601). But, since data has already been absent in the file, "absence" is determined in the decision process (step 602) and the program proceeds to the next process (step 604 in Fig. 7).--

Please replace the paragraph beginning at page 18, line 3, with the following rewritten paragraph:

--A segment 20 which is next in terms of time series is pointed to (step 705) and time information (March, 5 1994) is acquired from the bookmark information

area 22 of that segment 20. The acquired time information (March, 1994) is compared with February, 1994 for the deletion object and it is determined that the segment 20 is not the deletion object (step 701), thus ending the deletion processing. After the completion, the database assumes a state as shown in Fig. 11.--

Please replace the paragraph beginning at page 18, line 12, with the following rewritten paragraph:

--In the present deletion processing, internal data need not be directly accessed and only the bookmark information area is taken as the object, thereby making it possible to perform deletion within a short time and [during] while online.--

Please replace the paragraph beginning at page 18, line 17, with the following rewritten paragraph:

--The segments are used in wrap-around fashion to attain an advantage in that no reorganization is needed even when addition/deletion is repeated. Finally, the wrap-around architecture will be described.--

Please replace the paragraph beginning at page 19, line 22, with the following rewritten paragraph:

--As described above, according to the embodiments of the present invention, the intended data can be accessed without [resort] resorting to an index by retrieving thoroughly only the specified control information storage range without retrieving the whole of the database.--

FIG. 3

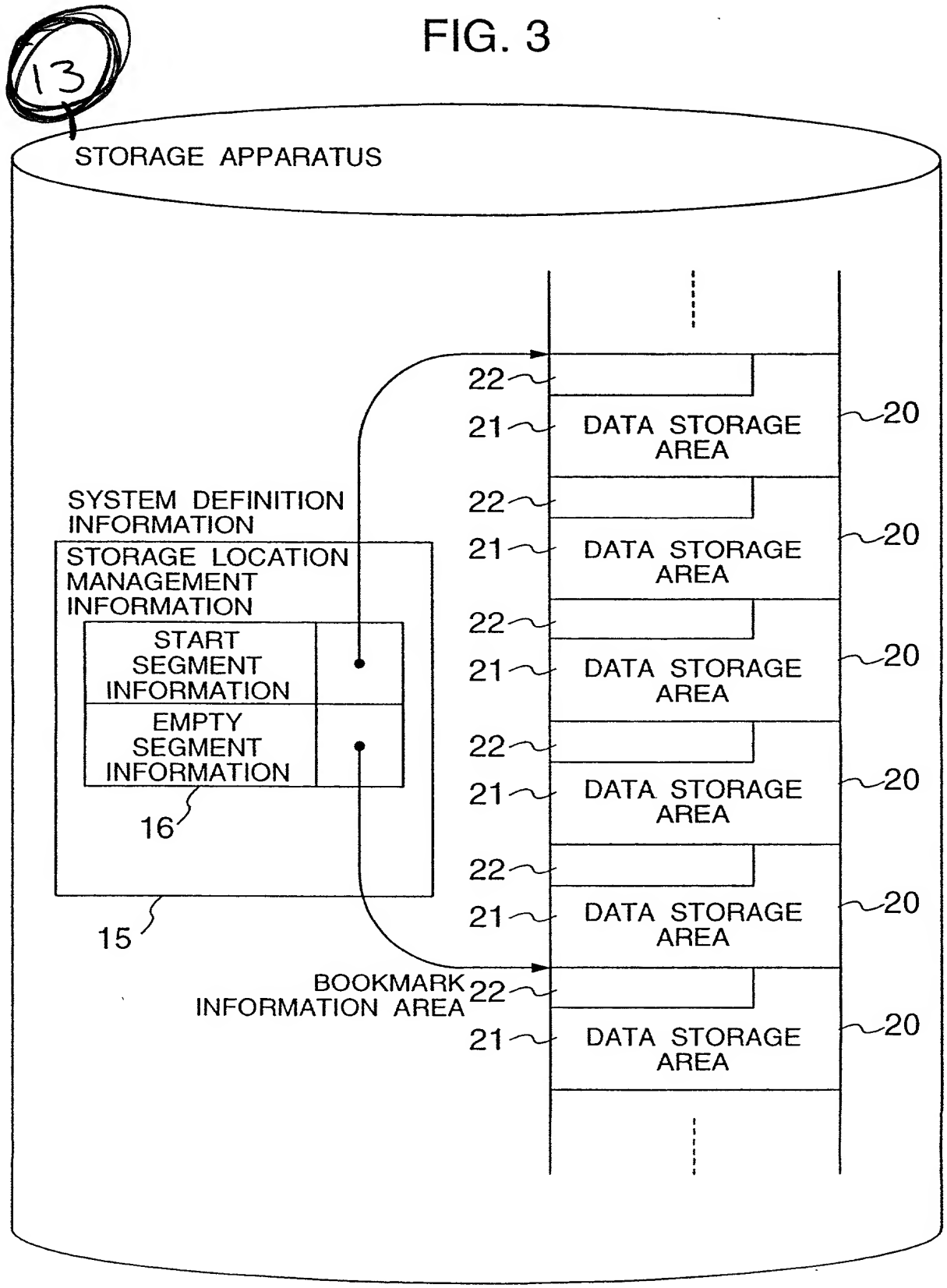

13
STORAGE APPARATUS
SYSTEM DEFINITION INFORMATION
STORAGE LOCATION MANAGEMENT INFORMATION
START SEGMENT INFORMATION
EMPTY SEGMENT INFORMATION
16
15
BOOKMARK INFORMATION AREA
22
21
DATA STORAGE AREA
20

FIG. 5

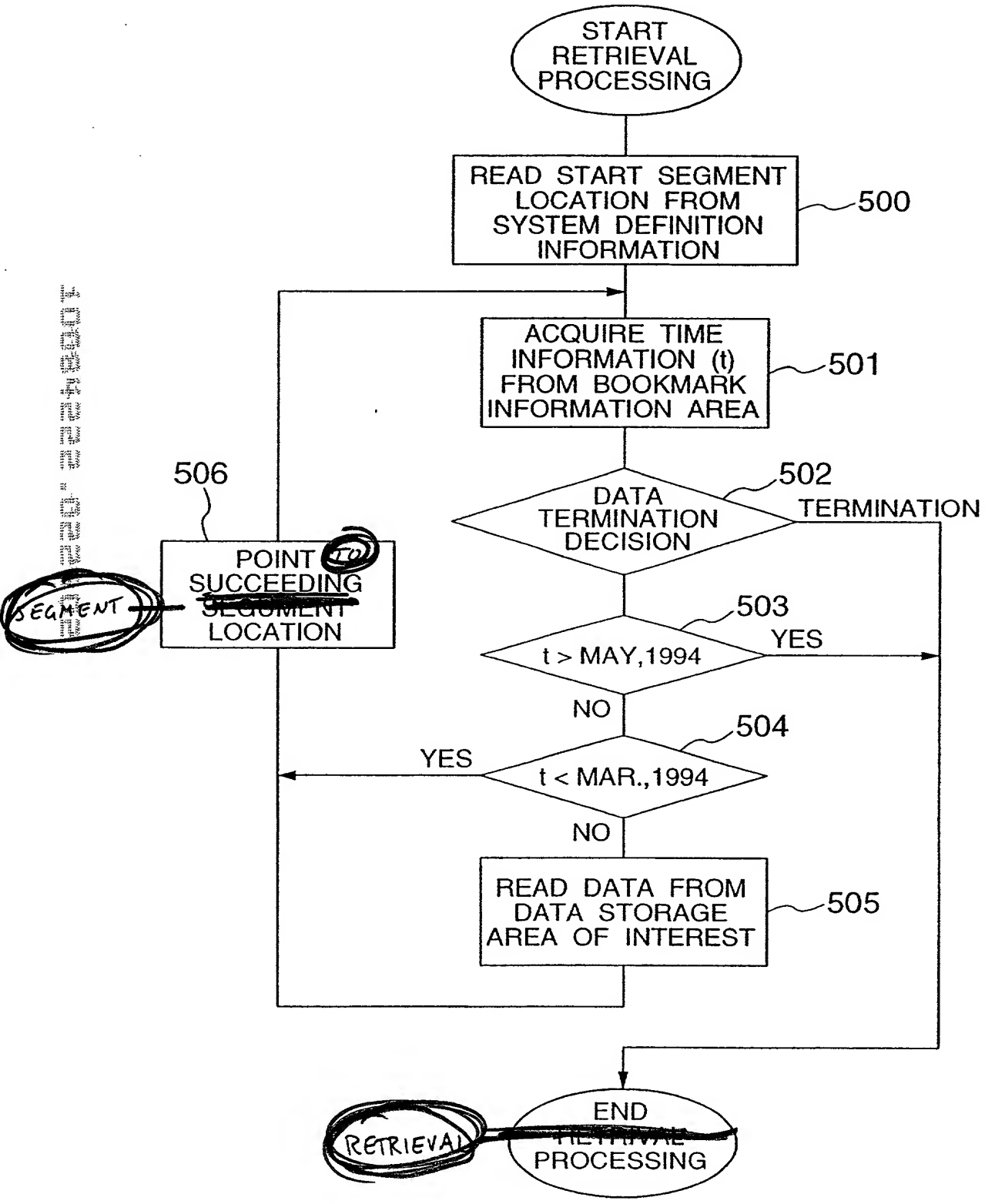

START RETRIEVAL PROCESSING
READ START SEGMENT LOCATION FROM SYSTEM DEFINITION INFORMATION
500
ACQUIRE TIME INFORMATION (t) FROM BOOKMARK INFORMATION AREA
501
DATA TERMINATION DECISION
502
TERMINATION
506
POINT TO SUCCEEDING SEGMENT LOCATION
t > MAY,1994
503
YES
NO
t < MAR.,1994
504
YES
NO
READ DATA FROM DATA STORAGE AREA OF INTEREST
505
END RETRIEVAL PROCESSING

FIG. 7

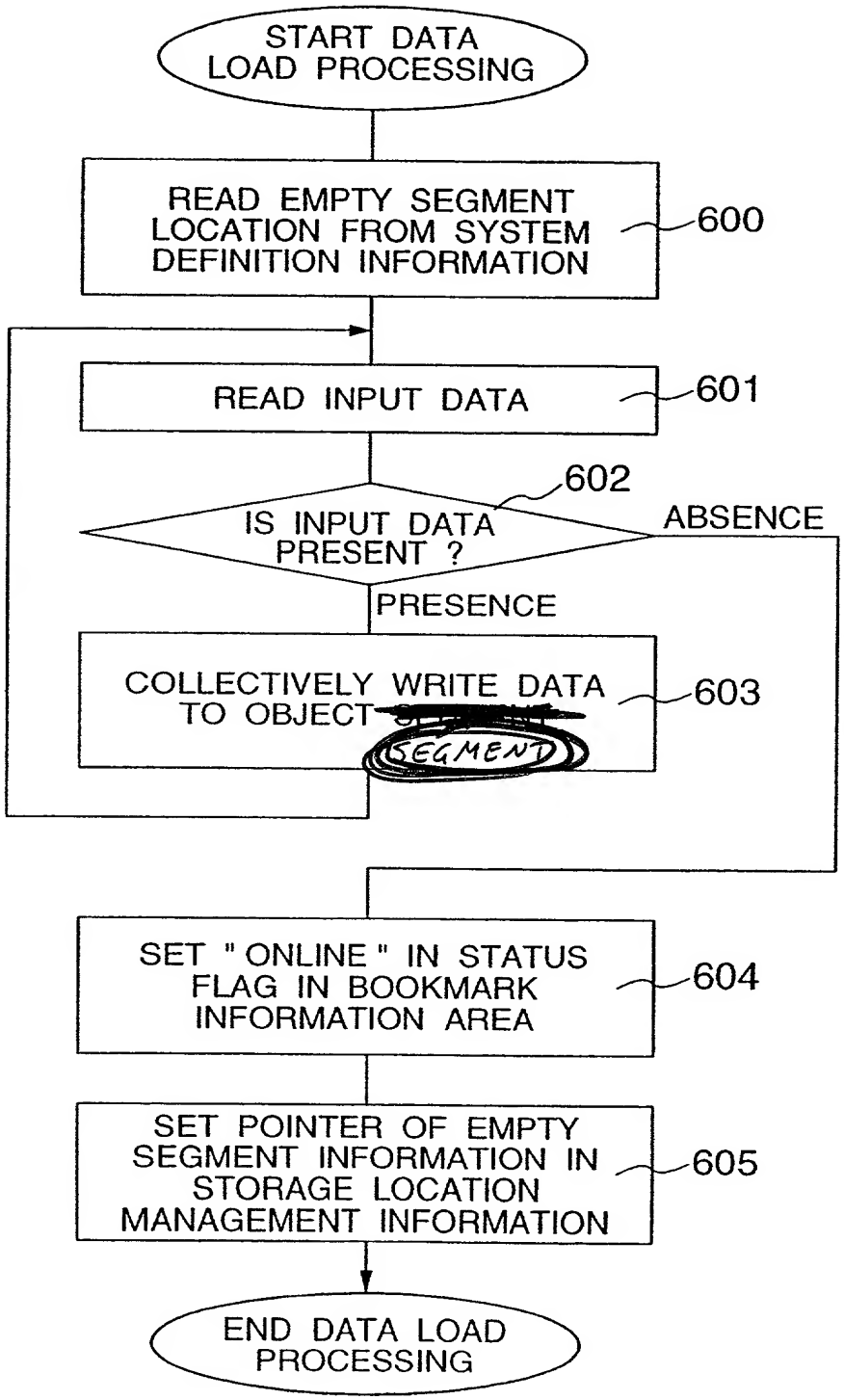
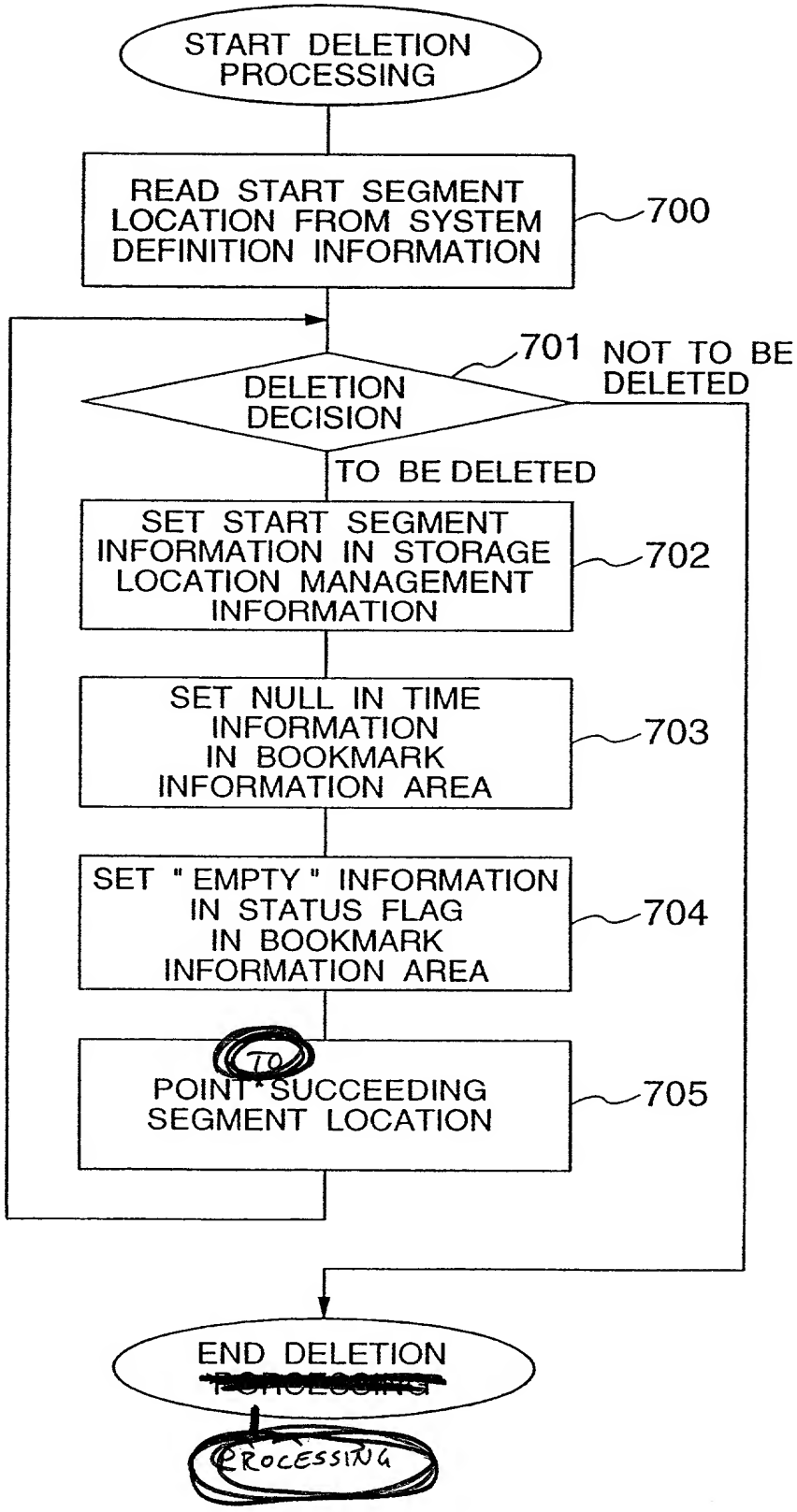

START DATA LOAD PROCESSING
READ EMPTY SEGMENT LOCATION FROM SYSTEM DEFINITION INFORMATION
600
READ INPUT DATA
601
602
IS INPUT DATA PRESENT ?
ABSENCE
PRESENCE
COLLECTIVELY WRITE DATA TO OBJECT SEGMENT
603
SET "ONLINE" IN STATUS FLAG IN BOOKMARK INFORMATION AREA
604
SET POINTER OF EMPTY SEGMENT INFORMATION IN STORAGE LOCATION MANAGEMENT INFORMATION
605
END DATA LOAD PROCESSING

FIG. 10


START DELETION PROCESSING
READ START SEGMENT LOCATION FROM SYSTEM DEFINITION INFORMATION
700
DELETION DECISION
701
NOT TO BE DELETED
TO BE DELETED
SET START SEGMENT INFORMATION IN STORAGE LOCATION MANAGEMENT INFORMATION
702
SET NULL IN TIME INFORMATION IN BOOKMARK INFORMATION AREA
703
SET "EMPTY" INFORMATION IN STATUS FLAG IN BOOKMARK INFORMATION AREA
704
TO POINT SUCCEEDING SEGMENT LOCATION
705
END DELETION PROCESSING